

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MODULY A FIRMWARE PRO IR KOMUNIKACI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID SLEZÁK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MODULY A FIRMWARE PRO IR KOMUNIKACI

MODULES AND FIRMWARE FOR IR COMMUNICATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID SLEZÁK

VEDOUcí PRÁCE

SUPERVISOR

Ing. JOSEF STRNADEL, Ph.D.

BRNO 2012

Abstrakt

Tato bakalářská práce se zabývá návrhem modulů komunikujících v infračervené oblasti. Rozebírá strukturu protokolu NEC, SIRC, RC5 a standardu IrDA. Dále popisuje návrh hardwarových modulů řízených mikroprocesory rodiny HC08, které komunikují těmito protokoly. Jádrem práce jsou obslužné podprogramy mikroprocesoru psané v jazyce assembler a ukázka jejich použití.

Abstract

This bachelor's thesis is about the design of modules communicating in the infrared region. It analyzes the structure of protocols NEC, SIRC, RC5 and IrDA standards. Describes the design of hardware modules controlled by a microprocessor HC08 family that communicate these protocols. The core of work are routines written in assembler language and a demonstration of their use.

Klíčová slova

NEC, SIRC, RC5, IrDA, SPI, infračervené záření, elektromagnetické vlnění, HC08, Nitron, Janus

Keywords

NEC, SIRC, RC5, IrDA, SPI, infrared radiation, electromagnetic waves, HC08, Nitron, Janus

Citace

David Slezák: Moduly a firmware pro IR komunikaci, bakalářská práce, Brno, FIT VUT v Brně, 2012

Moduly a firmware pro IR komunikaci

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Josefa Strnadela.

.....

David Slezák
10. května 2012

Poděkování

Děkuji panu Josefu Strnadelovi za poskytnutí literatury a za odborné rady a pomoc při tvorbě práce. Také děkuji za zapůjčení vývojového kitu a ostatních součástek.

© David Slezák, 2012.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Zadání bakalářské práce

Řešitel: **Slezák David**

Obor: Informační technologie

Téma: **Moduly a firmware pro IR komunikaci**
Modules and Firmware for IR Communication

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s platformou HC(S)08 firmy Freescale, s principy bezdrátové komunikace pomocí prvků pracujících v infračervené (IR) oblasti elektromagnetického spektra a s protokoly IrDA, RC5(X), NEC, SIRC.
2. Zvolte, popř. navrhnete přijímací a vysílací moduly umožňující spolehlivou IR komunikaci nad protokoly IrDA, RC5(X), NEC, SIRC a experimentálně ověřte základní funkčnost prototypů těchto modulů.
3. Implementujte firmware (tj. sadu funkcí) umožňujících vysílání a příjem dat s využitím IR modulů z předchozího bodu.
4. Navrhnete jednoduchou aplikaci využívající firmware z předchozího bodu, implementujte ji a vhodným způsobem demonstруйте její funkčnost v několika praktických oblastech (např. dálkové řízení spotřebičů, sledování stavu čidel).
5. Diskutujte výhody a nevýhody spojené s použitím aplikace v praxi a diskutujte možné způsoby jejich řešení.

Literatura:

- Infrared Data Association [online]. Dokument dostupný z <http://www.irda.org/>.
- Bergmans, S.: SB-Projects: IR remote control [online].
<http://sbprojects.com/knowledge/ir/>.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Strnadel Josef, Ing., Ph.D., UPSY FIT VUT**

Datum zadání: 1. listopadu 2011

Datum odevzdání: 16. května 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2



doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Obsah

1 Úvod	5
2 Standardy a protokoly pro komunikaci v oblasti infračerveného záření	6
2.1 Protokoly dálkových ovladačů	6
2.1.1 RC5 protokol	7
2.1.2 NEC Protokol	8
2.1.3 SIRC Protokol	9
2.2 Standardy organizace IrDA	10
2.2.1 Koncept podle IrDA	10
2.2.2 Fyzické vlastnosti přenosu	12
3 Návrh HW modulů pro IR komunikaci	13
3.1 Platforma HC08 firmy Freescale	13
3.1.1 Konkrétní mikroprocesor	14
3.1.2 Vývojový kit Janus	15
3.2 Návrh přenosové části modulů	15
3.2.1 Moduly protokolů dálkových ovladačů	16
3.2.2 Modul standardu IrDA	18
4 Implementace protokolů	19
4.1 RC5 protokol	19
4.2 NEC protokol	21
4.3 SIRC protokol	23
4.4 IrDA přenos	25
5 Ukázková aplikace	27
5.1 Popis a řešení ukázkové aplikace	27
5.1.1 Volba druhého rozhraní modulu	27
5.1.2 Návrh protokolu SPI rozhraní a implementace v HC08	28
5.2 Celková implementace aplikace	28
5.3 Zapojení	29
5.4 Použití aplikace	29
6 Dokumentace	30
6.1 Podprogramy protokolů dálkových ovladačů	30
6.2 Podprogramy standardu IrDA	30
7 Závěr	31

A	Obsah CD	34
B	Vývojové diagramy ukázkové aplikace	35
C	Fotografie praktické části	37

Seznam obrázků

2.1	Datový rámec RC5 protokolu	7
2.2	Kódování Manchester	7
2.3	Datový rámec NEC protokolu	8
2.4	Protokol NEC a) start sekvence b) logická 1 c) logická 0	9
2.5	Datový rámec SIRC protokolu	9
2.6	Protokol SIRC a) start sekvence b) logická 1 c) logická 0	10
2.7	Základní vrstvý model podle IrDA	10
2.8	Pulsni modulace používána v SIR, převzato z [7]	11
3.1	Blokové schéma TIM jednotky, převzato z [4]	14
3.2	Vývojový kit Janus, převzato z [11]	15
3.3	Principiální schéma zapojení fotodiody a led diody, čerpáno z [9]	16
3.4	Zapojení vysílače IR signálu	16
3.5	Zapojení přijímače IR signálu	17
3.6	Blokové schéma modulu HSDL-3612, převzato z katalogového listu [14], doporučené hodnoty externích součástek jsou: $c_1 = 470 \text{ nF}$ $c_2 = 6.8 \text{ }\mu\text{F}$ a $R_1 = 15 \text{ }\Omega$	18
4.1	Úsek kódu v jazyce assembler generující modulační frekvenci	20
4.2	Příjem RC5 protokolu	21
4.3	Příjem NEC protokolu	22
4.4	Příjem SIRC protokolu	24
4.5	Výstup generovaný obvodem HSDL-3612	26
5.1	Blokové schéma modulu ukázkové aplikace	28
B.1	Vývojový diagram řídicího programu Nitronu v ukázkové aplikaci	35
B.2	Vývojový diagram řídicího programu Fitkitu v ukázkové aplikaci	36
C.1	Zapojení ukázkové aplikace	37
C.2	Testování modulu pro IrDA komunikaci	38

Seznam tabulek

2.1	Příkon zařízení podle IrDA	12
2.2	Maximální vzdálenost zařízení podle IrDA	12
3.1	Parametry použité led diody z katalogového listu	16
3.2	Parametry obvodu PL-IRM0101-3, čerpáno z [5]	17
4.1	Zvolená tolerance přijímaných časových průběhu NEC protokolu. * jednotka takty značí počet hodinových pulsů čítače a její hodnota je v šestnáctkové číselné soustavě	23
4.2	Zvolená tolerance přijímaných časových průběhu SIRC protokolu. * jednotka takty značí počet hodinových pulsů čítače a její hodnota je v šestnáctkové číselné soustavě	25
5.1	Volba IR protokolů v ukázkové aplikaci	29
6.1	Hodnoty konstant odpovídající přenosovým rychlostem	30

Kapitola 1

Úvod

Bezdrátový přenos dat je v poslední době vyžadován u většiny zařízení. Byla vyvinuta řada technologií, které bezdrátový přenos dat umožňují. Využívají elektromagnetických vlnění různých vlnových délek. Tato práce se zabývá přenosem dat v oblasti elektromagnetického záření označovaného jako infračervená oblast. Nejširší aplikace technologie infračerveného přenosu je zastoupena v dálkovém ovládání spotřebičů. V dřívější době se využívala i pro přenosy větších objemu dat mezi zařízeními, dnes ji nahrazují technologie jako bluetooth nebo Wi-Fi.

Cílem práce je navrhnout fyzická zařízení a jejich firmware, které budou zajišťovat komunikaci v infračervené oblasti. Zaměřuje se na protokoly dálkových ovladačů NEC, SIRC, RC5 a standard IrDA. Cílovou platformou jsou mikroprocesory řady HC08 společnosti Freescale.

Druhá kapitola popisuje strukturu a kódování protokolů navržených firmami NEC, Philips, Sony a organizací IrAD. V minulosti si každá společnost vyvíjela vlastní protokoly a neexistoval žádný standardní, který by používalo více výrobců. V roce 1993 vznikla organizace IrDA a vytvořila stejnojmenný standard pro infračervenou komunikaci.

Fyzickou část komunikačních modulů popisuje třetí kapitola. Diskutuje výběr řídicího mikroprocesoru a součástek zajišťujících přenos paprsku.

Čtvrtá kapitola podrobněji rozebírá časové průběhy protokolů z implementačního pohledu. Jádrem kapitoly je popis implementace přijímačů a vysílačů jednotlivých protokolů v jazyce assembler.

Pátá kapitola ukazuje použití navržených vysílačů a přijímačů ve skutečné aplikaci. Navržené zařízení funguje jako převodník mezi SPI rozhraním a zvoleným infračerveným protokolem.

Kapitola Dokumentace je návod jak pracovat s podprogramy vytvořenými jako součást této práce. Popisuje způsob předávání parametrů jednotlivým podprogramům a příznaky, které nastavují. Kapitola dále zdůrazňuje zdroje mikroprocesoru, které patřičné podprogramy využívají a nelze je používat v konečné aplikaci.

V závěru práce diskutuji vhodnost použití vytvořených podprogramů a modulů v reálných aplikacích. Dále shrnuji dosažené cíle práce, poukazuji na nedostatky a jejich možná řešení.

Kapitola 2

Standardy a protokoly pro komunikaci v oblasti infračerveného záření

Oblast infračerveného záření poprvé objevil W. Herschelem v roce 1801. Je to část elektromagnetického spektra. Vlnové délky jsou větší než u viditelného světla a nacházejí se v intervalu 760 nm až 1000 nm. Záření způsobuje pohyb částic v tělesech s teplotou vyšší než je absolutní nula, podle [16]. Zdroj záření tedy vznikne dodáním energie. Mezi nejčastější zdroje záření patří tepelné, rádiové a luminiscenční.

V praxi se pro komunikaci v oblasti infračerveného záření, anglicky *infra red* (IR), používají luminiscenční LED diody, které jsou zdrojem elektromagnetického záření i v této oblasti, podle [15]. S LED diodou se pracuje pouze ve dvou stavech: vyzařujícím paprsek a nevyzařujícím. Pro snížení okolních vlivů rušení přenosového kanálu se používají různé modulace.

2.1 Protokoly dálkových ovladačů

Dálkové ovladače využívají frekvenční modulace s nosnou frekvencí okolo 30 až 60 KHz. Přenášený signál v konkrétním časovém úseku je reprezentován změnou stavů LED diody. Rozlišuje se stav s odpovídající modulační frekvencí a stav nevyzařující paprsek. První případ se v literatuře označuje jako značka, anglicky *mark*, druhý jako mezera, anglicky *space*.

Jednotlivé bity datového rámce, které se přenášejí, mají různý poměr značky a mezery vzhledem k bitové periodě. Závisí na zvoleném způsobu kódování logických hodnot.

Datové rámce jsou přenášeny asynchronně, bit po bitu. To znamená, že součástí přenosového kanálu není samostatný synchronizační signál. Používá se kódování dat, které zakóduje do datového rámce kromě dat i synchronizační signál. Označují se jako kódy se zabudovaným synchronizačním signálem anglicky *embedded clock*. Příjemce ho umí rekonstruovat nebo používá předem dohodnutý.

LED diody určené k vysílání paprsku obvykle vytvářejí záření o vlnové délce 950 nm. Elektrickou spotřebu diody lze ovlivnit změnou střídy modulační frekvence. U některých protokolů výrobce doporučuje konkrétní hodnotu, při níž je ještě zaručené spolehlivé doručování datových rámců.

Protokoly určené k dálkovému ovládání zařízení se vyznačují poměrně dlouhými pro-

dlevami mezi odesíláním jednotlivých datových rámců. Komunikaci inicializuje převážně člověk, který komunikuje pomaleji než stroj.

Veškeré časové průběhy protokolů v této kapitole budou popsány z pohledu vysílacích zařízení. Na přijímači budou značky a mezery invertovány viz podkapitola 3.2.1.

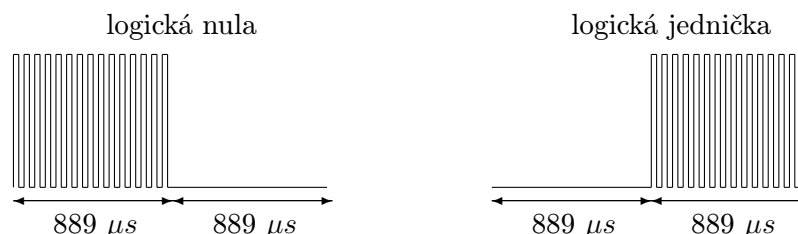
2.1.1 RC5 protokol

Protokol RC5 byl navržen firmou Philips pro dálkové ovládání spotřebičů a vestavěných zařízení. Často je implementován v dálkových ovládacích televizních přijímačů. Podle [2] datový rámec má velikost 14 bitů a skládá se ze tří částí viz obr.2.1:

- **Hlavička** obsahuje dva start bity a jeden **toggle bit**, jímž lze detekovat příjem totožných rámců (delší doba stisknutí tlačítka na ovladači).
- **Adresa zařízení** je pětibitová, což umožňuje adresovat až $2^5 = 32$ zařízení komunikujících protokolem RC5. Firma Philips vytvořila standard, přiřazující určité adresy konkrétním zařízením. Například televizní přijímače mají adresy nula a jedna. Adresa se přenáší od nejvíce významného bitu.
- **Data** jsou šestibitová. Interpretace této části rámce, podle zmiňovaného standardu, závisí na konkrétnějším zařízení (rozlišeno podle adres). Jsou přenášena od nejvíce významného bitu.

hlavička (3 bity)	adresa (5 bitů)	data (6 bitů)
-------------------	-----------------	---------------

Obrázek 2.1: Datový rámec RC5 protokolu



Obrázek 2.2: Kódování Manchester

Datový rámec je kódován kódem **Manchester**, který je **dvoufázový**. Dvakrát za periodu bitu mění polaritu signálu. Kodér lze realizovat exkluzivním logickým součinem kódovaného signálu s hodinovým signálem o dvakrát menší periodě než je bitová perioda kódovaného signálu. Na obrázku 2.2 je vidět zakódované logické hodnoty a namodulované na nosnou frekvenci 36 *KHz*. Počet period nosné frekvence za periodu jednoho bitu je dána jejich součinem:

$$1,778 * 36 \doteq 64.$$

Celková doba potřebná k přenesení jednoho rámce je dána počtem bitů vynásobených bitovou periodou:

$$889 * 2 * (3 + 5 + 6) = 24,892 \text{ [ms]}.$$

Podle [13] je další datový rámce odeslán nejdříve za 114 *ms*, od doby odeslání prvního bitu rámce. Pokud aplikace bude rozlišovat jednorázový stisk tlačítka a delší stisk, je třeba brát v úvahu toto zpoždění.

Z důvodu snížení spotřeby firma Philips doporučuje zvolit střidu paprsku vysílací LED diody na 1/3 až 1/4.

2.1.2 NEC Protokol

Protokol NEC vyvinula stejnojmenná firma, pro dálkové ovládače. Datový rámec podle [1] je na obrázku 2.3. Skládá se ze tří hlavních částí:

- **Start sekvence** tvoří jakousi hlavičku rámce viz obrázek 2.4. Nelze zde hovořit o počtu bitů vzhledem k následujícím částem datového rámce, protože tyto části mají bity jinak zakódované.
- **Adresa** zařízení je šestnáctibitová. Skládá ze dvou osmibitových částí. První je skutečná adresa a druhá je její bitová negace. Důvodem je detekce chyb v přenosovém kanále.
Adresa je přenášena od nejméně významného bitu.
- **Data** jsou přenášena stejným způsobem jako adresa a mají i stejný počet bitů. Podobně jako u RC5 protokolu, viz 2.1.1, byl vytvořen standard přiřazující jednotlivým bitovým kombinacím přenášných dat jednoznačný příkaz pro koncové zařízení. Počet kombinací je zde čtyřnásobný oproti RC5 protokolu: $2^8 = 256$.
Data se přenáší od nejméně významného bitu.

start sekvence	adresa (8 bitů)	\overline{adresa} (8 bitů)	data (8 bitů)	\overline{data} (8 bitů)
----------------	-----------------	------------------------------	---------------	----------------------------

Obrázek 2.3: Datový rámec NEC protokolu

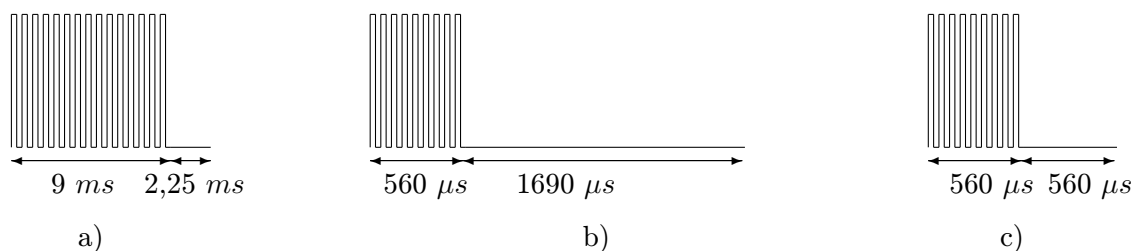
Protokol NEC pro kódování dat v přenosovém kanále používá **pulsní šířkovou modulaci**, obrázek 2.4. Ta určuje střidu, což je poměr mezi značkou a mezerou v rámci bitové periody. Logická jednička má střidu 1 : 4 ve prospěch mezery, logická nula 1 : 1. Rozdílná je i bitová perioda logických hodnot: logická jednička trvá 2,25 *ms* a logická nula 1,12 *ms*. Zakódované rámce jsou modulovány frekvenční modulací na nosný kmitočet 38 *KHz*.

Mechanismus detekce delšího držení stejného tlačítka na ovladači se provádí zasláním jednoho datového rámce a po 110 *ms*, od začátku přenosu start sekvence, zasláním pouze další start sekvence.

Celkovou dobu přenosu datového rámce lze vypočítat i přes různou bitovou periodu logických hodnot díky negovaným hodnotám adresy a dat. Celkem se přenáší 16 bitů a 16 jejich negací. V datovém rámci bude vždy 16 jedniček a 16 nul plus doba start sekvence. Vypočet je následující:

$$16 * 2,25 + 16 * 1,12 + 9 + 4,5 = 67,42 [ms].$$

Rozšíření tohoto protokolu nabízí zvětšení adresového prostoru zařízení na 16 bitů.



Obrázek 2.4: Protokol NEC a) start sekvence b) logická 1 c) logická 0

2.1.3 SIRC Protokol

Protokol SIRC navrhla firma Sony. Existují tři jeho modifikace, které se liší počtem přenášených bitů. V tomto textu se zaměřím na dvanáctibitovou verzi. Další dvě jsou patnáctibitové a dvacetibitové. Dvanáctibitová verze je popsána v [3] a její datový rámec je znázorněn na obrázku 2.5. Jednotlivé části jsou podobné jako v předchozích protokolech. Hlavní rozdíl spočívá v záměně pořadí adresové a datové části:

- **Start sekvence** má odlišnou střihu i periodu od obou logických hodnot používaných v dalších částech rámce, jak je zřejmé z obrázku 2.6. Střída je 1 : 5 ve prospěch značky a perioda má délku 3 ms.
- **Data** jsou přenášeny dříve než adresa. Jsou sedmi bitová a přenáší se nejdříve nejméně významný bit.
- **Adresa** představuje prostor $2^5 = 32$ možných zařízení. Stejně jako u dat je přenášen nejdříve nejméně významný bit.

start sekvence	data (7 bitů)	adresa (5 bitů)
----------------	---------------	-----------------

Obrázek 2.5: Datový rámec SIRC protokolu

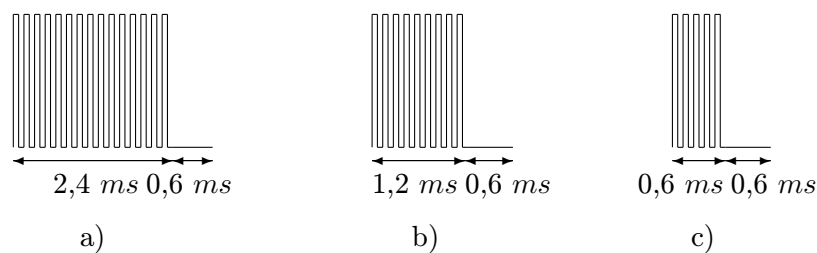
Jako u předchozích dvou protokolů byl vytvořen standard určující konkrétním typům zařízení, konkrétní adresy. Podobně byly datům přiřazeny jednoznačné příkazy.

SIRC protokol kóduje logické hodnoty pomocí pulsně šířkové modulace, viz obrázek 2.6. Střída pro logickou jedničku je 1 : 3 ve prospěch značky, při bitové periodě 1,8 ms. U logické nuly je střída 1 : 1 při periodě 1,2 ms. Kódovaná data jsou modulována na nosnou frekvenci 40 KHz frekvenční modulací.

Z důvodu proměnlivé délky bitových period u logických hodnot, nelze předem jednoznačně stanovit délku datového rámce. Můžeme stanovit pouze maximální možnou délku a to tak, že budeme předpokládat adresu i data ze samých logických jedniček. Výpočet bude vypadat následovně:

$$12 * 1,8 + 3 = 24,6 [ms].$$

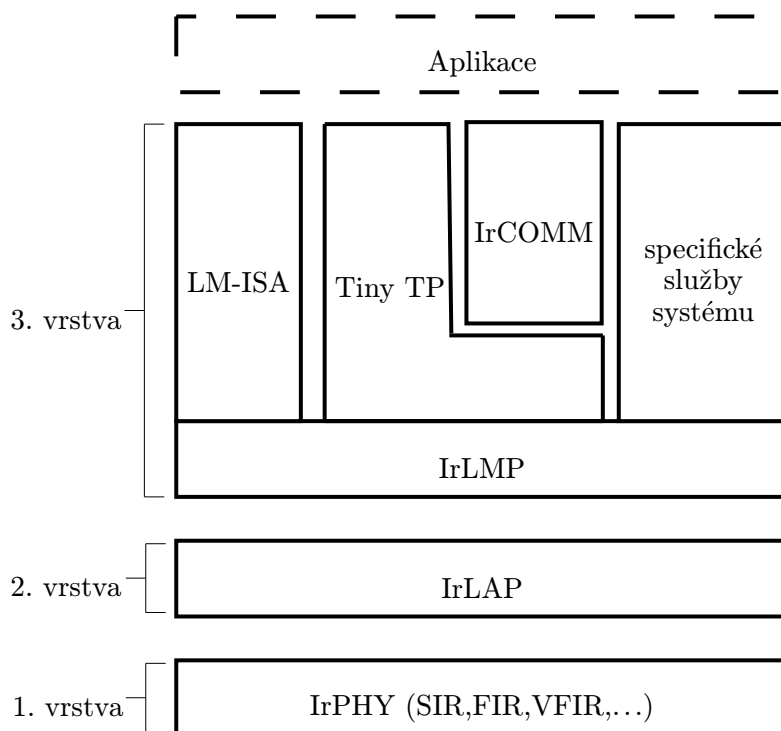
Podle [12] se přidržení tlačítka dálkového ovladače detekuje po přijetí tří stejných datových rámců. Přičemž datové rámce se mohou odesílat nejdříve 45 ms od doby obeslání start sekvence.



Obrázek 2.6: Protokol SIRC a) start sekvence b) logická 1 c) logická 0

2.2 Standardy organizace IrDA

Organizace **Infrared Data Association**, zkráceně **IrDA**, je nezisková organizace, která vytváří a spravuje standardy pro sériový přenos dat v infračervené oblasti, viz [8]. Standard z roku 1993 definuje základní koncepci, viz [6], která se stále používá. Tvoří třívrstvý model, je zobrazen na obrázku 2.7. Každá vrstva předává data se svoji hlavičkou nižší vrstvě, stejně jako v referenčním ISO/OSI modelu počítačové sítě.



Obrázek 2.7: Základní vrstvý model podle IrDA

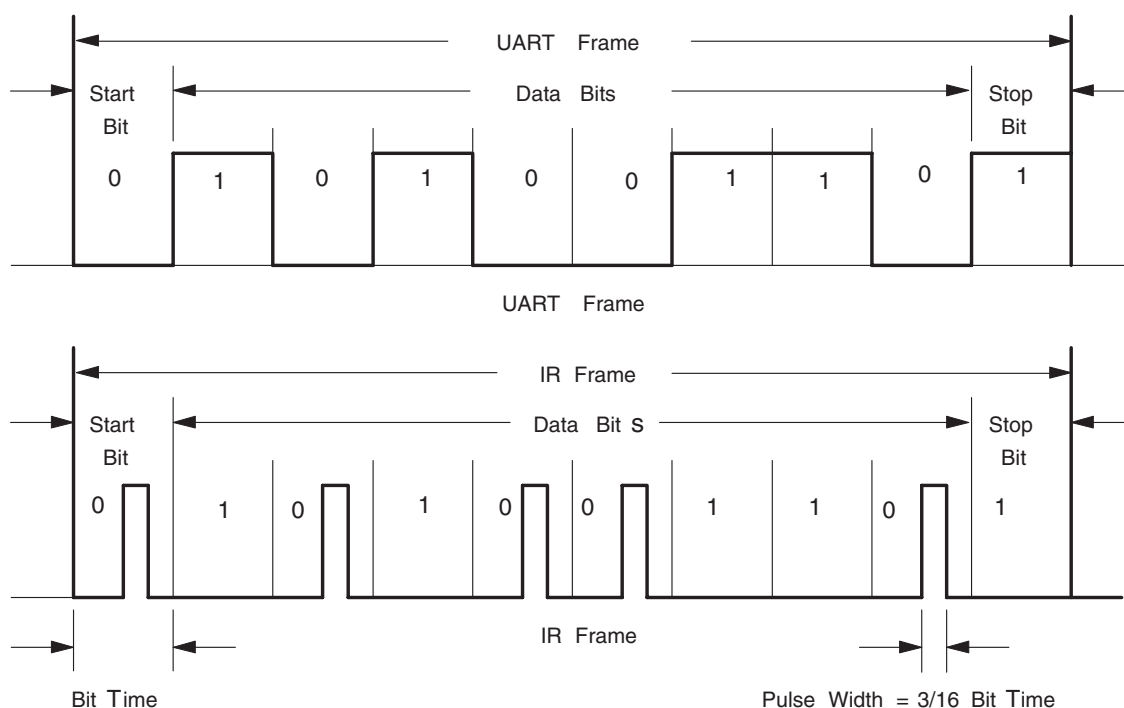
2.2.1 Koncept podle IrDA

Následuje popis protokolů na jednotlivých vrstvách:

- **IrPHY** přenáší data na nejnižší úrovni, bit po bitu. Používá asynchronní přenos

totožný s přenosem UART, tj. jeden start bit, osm datových bitů a jeden stop bit. Standard definuje několik protokolů na této vrstvě, podle přenosové rychlosti a způsobu modulace dat:

- SIR (serial IR) komunikuje maximální rychlostí $115,2 \text{ Kbit/s}$ a je použita pulsní modulace dat. Logické nule odpovídá střída 3/16 ve prospěch mezery, logické jedničky mezera po celou dobu bitové periody, viz obrázek 2.8.
- FIR (fast IR) rychlost až 4 Mbit/s s PPM (pulse-position modulation) modulací dat.
- VFIR (very fast IR) rychlost až 16 Mbit/s .



Obrázek 2.8: Pulsní modulace používána v SIR, převzato z [7]

- **IrLAP** (Link Access Protocol) je odvozen od asynchronního komunikačního protokolu HDLC používaného v počítačových sítích, kde pracuje na linkové vrstvě v referenčním IOS/OSI modelu. Služby protokolu jsou upraveny pro potřeby IR přenosu. Mezi základní patří: detekce ostatních zařízení, kontrola kolize jejich adres, vytvoření spojení mezi nimi a nastavení přenosových parametrů jako rychlost a velikost dat atd. V komunikaci je vždy pouze jedno zařízení, které řídí komunikaci. Může současně komunikovat i s více zařízeními.
- **IrLMP** (Link Management Protocol) detekuje služby poskytované připojenými zařízeními a umožňuje mezi nimi přepínat. V modelu ISO/OSI by protokol pracoval na

síťové vrstvě.

Nad IrLMP protokolem na 3. vrstvě je definována řada dalších protokolů zajišťujících specifické služby:

- LM-ISA (Link Management Information Access Service) spravuje informace o dostupných službách zařízení. Jedná se o podpůrný protokol IrLMP.
- Tiny TP (Tiny Transport Protocol) umožňuje segmentaci dat a tím přenášet větší zprávy než je maximální dohodnutá velikost přenášených dat při ustavení spojení na 2. vrstvě. Provádí kontrolu toku jako je správné doručování ve správném pořadí.
- IrCOMM emuluje sériový a paralelní port počítače.
Jinou možností komunikace je pomocí vrstvy IrPHY přímo posílat data bez účasti protokolů ostatních vrstev. Potom v jednom okamžiku lze realizovat přenos pouze jedním směrem.
- Další protokoly jsou pro specifické aplikace jako je emulace počítačové sítě, emulace USB portu a další. Většina z nich využívá služeb Tiny TP protokolu.

2.2.2 Fyzické vlastnosti přenosu

Spolehlivé přenášení informací pomocí IrDA standardů vyžaduje dodržovat určitá pravidla, popsána v [7]:

- Zařízení jsou rozdělena na standardní a nízkopříkonová viz tabulka 2.1.

Typ zařízení	IrPHY protokol	Rozsah	jednotka
nízkopříkonové	SIR	3,6 - 72	mW/sr
	FIR a VFIR	9 - 72	mW/sr
standardní	SIR	40 - 500	mW/sr
	FIR a VFIR	100 - 500	mW/sr

Tabulka 2.1: Příkon zařízení podle IrDA

- Maximální vzdálenost dvou komunikujících zařízení je závislá na typu jejich spotřeby, jak ukazuje tabulka 2.2.

nízkopříkonové - nízkopříkonové	0,2 m
nízkopříkonové - standardní	0,3 m
standardní - standardní	1 m

Tabulka 2.2: Maximální vzdálenost zařízení podle IrDA

- Úhel mezi přijímačem a vysílačem musí být v rozsahu 15° až 30°.

Kapitola 3

Návrh HW modulů pro IR komunikaci

Kapitola popisuje výběr elektronických součástek modulů s ohledem na cenu a optimální využití jejich potenciálu. Cílem je navrhnout levné, konstrukčně nenáročné ale spolehlivé moduly pro IR přenos.

3.1 Platforma HC08 firmy Freescale

V roce 2004 došlo k oddělení polovodičové divize společnosti **Motorola** a vznikla nová společnost **Freescale**. Procesory rodiny HC08 byly vytvořeny ale ještě před tímto rozdělením. Dále v textu bude jmenována pouze společnost Freescale, jelikož firma Motorola se touto problematikou již nezabývá.

Charakteristické vlastnosti rodiny HC08 byly získány z [10]. Jádrem HC08 bylo navrženo pro konstrukci levných, rychlých mikroprocesorů s nízkou spotřebou. Je označováno za osmibitové, disponuje ale šestnáctibitovou adresní sběrnici a několika šestnáctibitovými registry. Díky tomu může adresovat paměťový prostor o velikosti 64 KB. Architektura je typu von Neumann. Nabízí šestnáct adresovacích módů a až dva přístupy do paměti v jedné instrukci. Periférie jsou mapovány do paměti. Mikroprocesory s tímto jádrem disponují Flash pamětí. ALU jednotka je osmibitová a poskytuje hardwarové násobení, dělení, sečítání s přenosem i bez.

Po resetu může mikroprocesor jádra HC08 pracovat ve dvou módech: **user** a **monitor**. Druhý ze jmenovaných umožňuje programovat a ladit program přímo v aplikaci pomocí jediného pinu tzv. In Circuit Development zkráceně ICD. V User modu jsou sekvenčně prováděny instrukce uložené v paměti.

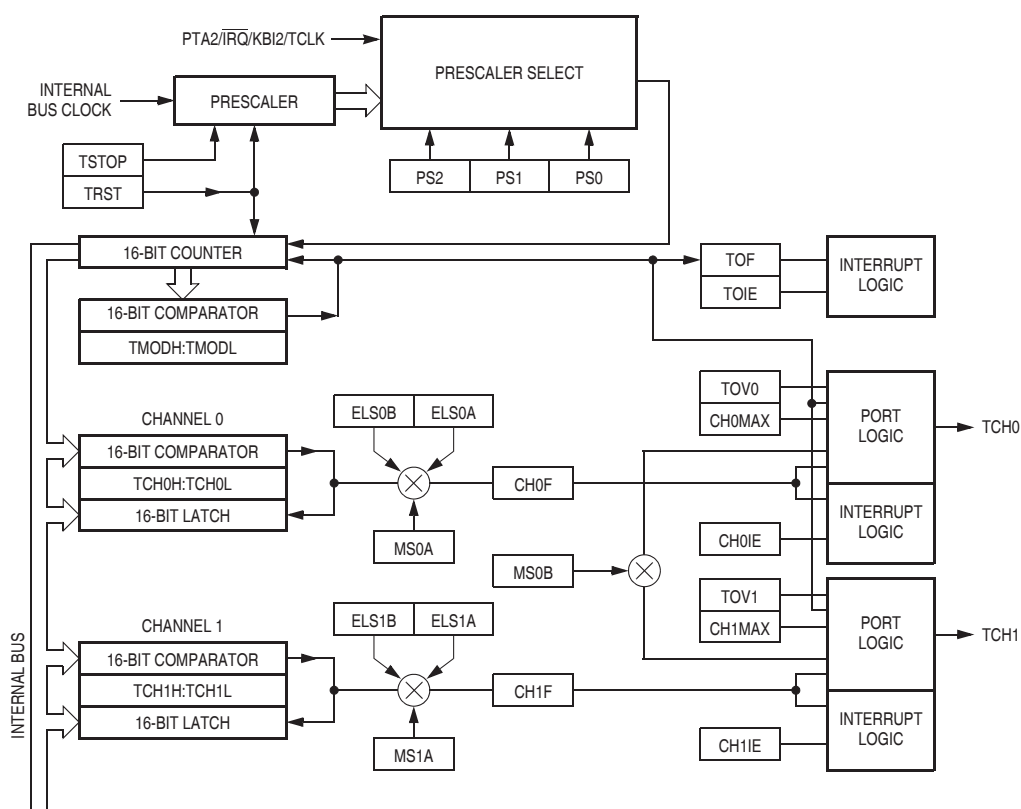
Programový model je navržen pro podporu vyšších programovacích jazyků. Skládá se ze tří šestnáctibitových registrů: indexovací, ukazatel zásobníku, programový čítač a ze dvou osmibitových: akumulátor a příznakový registr.

Procesory jsou dále vybaveny řadou periferních obvodů. Kromě zmiňovaného ICD, poskytuje A/D převodník, vnitřní zdroj hodinového signálu, jednotky pro detekci hran signálu, dvoukanálový šestnáctibitový čítač/časovač a další. Přesnost vnitřního zdroje hodin lze kalibrovat až na přesnost 5%, což většině aplikací postačuje. Současně to snižuje počet externích součástek konečné aplikace.

3.1.1 Konkrétní mikroprocesor

Na doporučení vedoucího bakalářské práce jsem vybral mikroprocesor MC68HC908QY4. Mezi jeho hlavní výhody patří nízká cena a pouzdro s poměrně málo piny, celkem šestnáct. Podle označení lze určit některé klíčové vlastnosti obvodu. Rozhodující je především velikost flash paměti, typové značení řady a sada integrovaných periferních zařízení na čipu. Mikroprocesor obsahuje 4 KB flash paměti, je z řady HC08 a obsahuje TIM jednotku, která je klíčová pro tuto aplikaci. Její blokové schéma je na obrázku 3.1. Základní část tvoří šestnáctibitový volně běžící čítač. Umožňuje čítat takty vnitřní sběrnice s možností aktivování předděličky. V šestnáctibitovém modulo registru se nastaví maximální hodnota čítače. Po jejím dosažení mikroprocesor nastaví bit přetečení čítače a může vyvolat přerušení, je-li povoleno. TIM jednotka disponuje dvěma kanály, číslovanými od nuly. Oba pracují s vlastním šestnáctibitovým komparačním registrem. Každý kanál může pracovat v režimu zachytu hrany nebo jako běžný čítač. Režim zachytu hrany umožňuje detekovat zachytnou hranu, sestupnou nebo obě hrany najednou. Hrana se detekuje na nultém a prvním pinu portu A, podle zvoleného kanálu. Po detekování hrany se nastaví příslušný příznak a může se vyvolat přerušení a do komparačního registru se nakopíruje hodnota volně běžícího čítače.

Druhá konfigurace kanálu vyvolá přerušení pokud se hodnota komparačního registru shoduje s hodnotou volně běžícího čítače. Současně s přerušením lze nastavit automatické negování příslušného pinu portu A.



Obrázek 3.1: Blokové schéma TIM jednotky, převzato z [4]

3.1.2 Vývojový kit Janus

Jako vývojový kit jsem se rozhodl, také na doporučení vedoucího bakalářské práce, použít vývojový kit Janus viz obrázek 3.2. Kit byl vytvořen firmou Freescale pro snadné seznámení se s procesory Nitron, jak je také označována řada 68HC908. Obsahuje Can9 konektor pro připojení k sériovému portu počítače za účelem programování a ladění. Převod mezi standardem RS232 a TTL realizuje převodník MAX232 firmy Maxim. Kit napájen devíti volty, které tvoří vstupní napětí stabilizačního obvodu pro mikroprocesor. Na desce kitu je k mikroprocesoru připojen krystal o frekvenci 9,8304 MHz, který zajišťuje spolehlivý a přesný zdroj hodinového signálu mikroprocesoru.

Kromě resetovacího tlačítka a tlačítka přepínajícího mezi usr a monitor módy obsahuje ledky, tlačítka a trimr připojené na piny procesoru jako externí periférie.



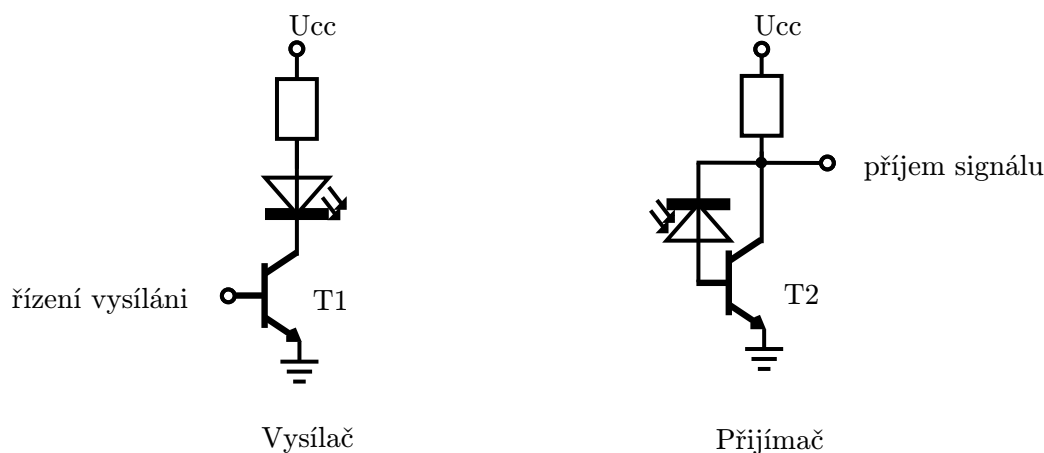
Obrázek 3.2: Vývojový kit Janus, převzato z [11]

3.2 Návrh přenosové části modulů

Přenos informace, ve formě elektrické energie, pomocí světelného záření lze realizovat dvojicí polovodičových elektronických součástek jako je fotodiody a LED dioda. Prochází-li LED diodou elektrický proud, vyzařuje světelnou energii ve formě fotonů. Naopak fotodiody vytváří po dopadu fotonů na její PN přechod elektrický proud. Kombinací obou součástek vznikne světelný přenosový kanál, kde LED dioda tvoří vysílač a fotodiody přijímač. Obrázek 3.3 ukazuje principiální zapojení s tranzistory: T1 řídí vysílání světelného paprsku a T2 pracuje jako zesilovač fotonového proudu fotodiody na přijímači.

Důležitou vlastností fotodiody je rychlost reakce na změnu intenzity osvětlení. Běžné fotodiody s obyčejným PN přechodem reagují v časech 10^{-6} s až 10^{-9} s. Vložením polovodiče s vlastní vodivostí mezi vrstvy P a N se dosahuje časů řádově tisíckrát nižších. Takové LED diody jsou označovány jako fotodiody PIN. Využívají se hlavně v oblasti IR přenosu.

Při výběru konkrétních součástek je třeba dbát na stejné vlnové délky vysílačů a přijímačů součástek.



Obrázek 3.3: Principiální schéma zapojení fotodiody a led diody, čerpáno z [9]

3.2.1 Moduly protokolů dálkových ovladačů

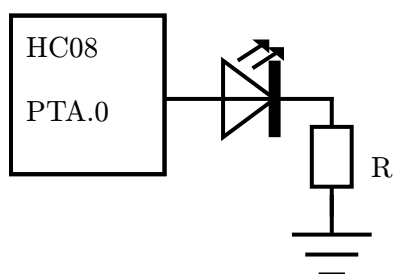
Fyzická část vysílače se skládá z led diody pracující v infračervené oblasti a ochranného odporu. Anoda led diody je připojená na 13. pin mikroprocesoru. Tento pin je nultý bit portu A. Využívá ho vnitřní periférie TIM 0. kanál. Tabulka 3.1 shrnuje důležité parametry zvolené led diody.

parametr	hodnota	jednotka
vlnová délka	940	nm
napětí na diodě	1,4	V
proud diodou	20	mA
vyzařovací úhel	20	°

Tabulka 3.1: Parametry použité led diody z katalogového listu

Ochranný odpor jsem zvolil podle výstupní úrovně logické jedničky procesoru, proud anapětí na led diodě, zapojení je na obrázku 3.4.

$$R = \frac{U_{log1} - U_{led}}{I_{led}} = \frac{5 - 1,4}{0,02} = 180 \text{ } \Omega.$$



Obrázek 3.4: Zapojení vysílače IR signálu

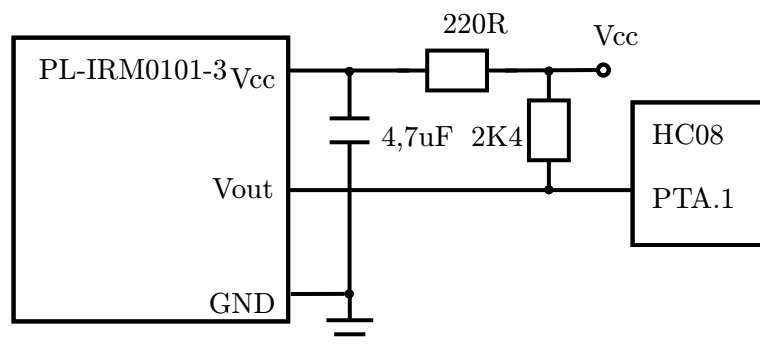
Přijímací část modulu je složitější než vysílací a nejde ji jednoduše sestavit z diskretních součástek. Proto řada výrobců nabízí integrované obvody, které umožňují demodulaci IR signálu s minimálním počtem diskretních součástek. Obsahují přijímač IR signálu se zesilovačem (princip viz obrázek 3.3) a demodulátor. Demodulátor se skládá z pásmové propusti a dekodéru. Pásmová propust filtruje rušení nebo signál modulovaný jinou frekvencí než pro jakou je obvod určen. Dekodér rozlišuje dva stavy filtrovaného signálu: značku a mezeru. Při detekci mezery generuje na výstupu vysokou logickou úroveň a při značce nízkou. Výstup z obvodu v závislosti na čase má invertovaný průběh oproti ještě nemodulovanému vysílanému signálu.

Každý z popisovaných protokolů v kapitole 2.1 je modulován na jiný nosný kmitočet. To by vyžadovalo demodulátor, pracující s patřičnou frekvencí, pro každý z nich. Někteří výrobci uvádějí u svých produktů seznam protokolů, které jimi budou správně demodulovány. Přitom uváděná demodulační frekvence obvodů je často jiná než u některých vyjmenovaných protokolů. Na základě těchto informací jsem se rozhodl vyzkoušet pro příjem všech protokolů integrovaný demodulátor PL-IRM0101-3 firmy HB Electronic Components, katalogový list dostupný z [5]. Výrobce garantuje spolehlivé dekódování protokolů SIRC, NEC i RC5. Pro testování jsem obvod zapojil dle pokynu výrobce a zkusil demodulovat signály s nosnými frekvencemi zmiňovaných protokolů: 36, 38 a 40 KHz. Všechny frekvence obvod demoduloval správně, proto jsem modul zhodnotil jako vhodný pro tuto práci a použil ho. Tabulka 3.2 ukazuje jeho klíčové vlastnosti.

parametr	hodnota	jednotka
vlnová délka	940	nm
pracovní frekvence	38	KHz
napájecí napětí	5,5	V
typ fotodiody	PIN	
podpora IR protokolů	NEC,SIRC,RC5	

Tabulka 3.2: Parametry obvodu PL-IRM0101-3, čerpáno z [5]

Zapojení přijímacího modulu ukazuje obrázek 3.5. K procesoru je připojen na pin 12. Odpovídá TIM jednotce 1. kanálu. Bude konfigurován jako detekce hrany.



Obrázek 3.5: Zapojení přijímače IR signálu

3.2.2 Modul standardu IrDA

Na trhu existuje množství integrovaných obvodů, které realizují přenos standardem IrDA na různých vrstvách a protokolech. Mezi jejich hlavní výhody patří malé rozměry, a minimální počet externích součástek. S počtem podporovaných protokolů samozřejmě roste i jejich cena.

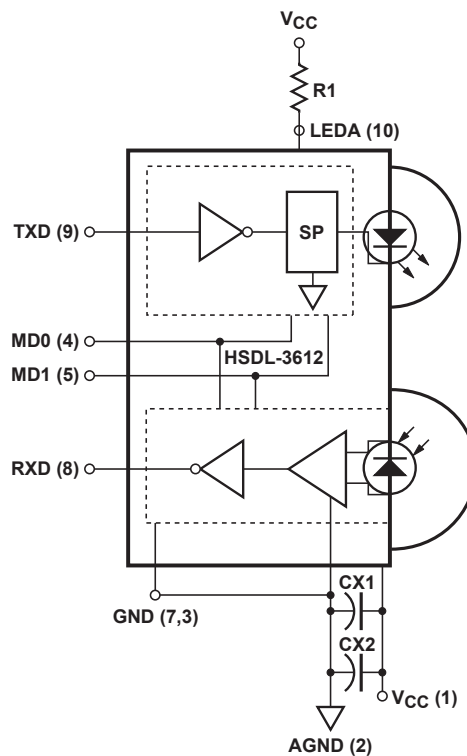
Rozhodl jsem se použít modul firmy Agilent Technologies: HSDL-3612. Jeho blokové schéma je na obrázku 3.6. Zajišťuje služby fyzické vrstvy IrDA standardu, protokolu SIR. Přenos je poloduplexní s přenosovou rychlostí 2,4 K b/s až 115,2 K b/s .

Pomocí kombinace logických hodnot na pinech 5 a 6 lze snížit jeho spotřebu během přenosu nebo ho úplně vypnout. V tomto režimu má nejnižší odběr proudu, ale není možné provádět přenosy dat.

Vysílací LED dioda vyzařuje paprsek o vlnové délce 870 nm. Přijímací optočlen je fotodioda typu PIN.

K mikroprocesoru řady HC08 jsem připojil pouze vysílací a přijímací piny modulu. Konkrétně pin TxD je připojen v PTA1 a RxD k PTA0. Pro správný příjem je nutno ještě mezi pin RxD a pin PTA0 připojit odpor o velikosti 34,5 K Ω .

V aplikaci, která vyžaduje nízkou spotřebu, je možno připojit zmiňované piny 5 a 6.



Obrázek 3.6: Blokové schéma modulu HSDL-3612, převzato z katalogového listu [14], doporučené hodnoty externích součástek jsou: $c_1 = 470 \text{ nF}$ $c_2 = 6.8 \text{ }\mu\text{F}$ a $R_1 = 15 \text{ }\Omega$

Kapitola 4

Implementace protokolů

Následující kapitola diskutuje několik možných řešení implementace protokolů. Konkrétněji je pak popsána implementace jednoho zvoleného způsobu.

Kvalitní přenos dat prostřednictvím popisovaných protokolů v kapitole 2 je závislý na dodržení časových průběhů. Časové průběhy budou dodrženy pokud budu mít k dispozici přesný spolehlivý zdroj hodinového signálu a čas prováděných režijních instrukcí bude dostatečně malý. První požadavek byl splněn při volbě komunikačních modulů v kapitole 3.1.2, konkrétně připojení krystalu. Pro splnění druhého požadavku jsem se rozhodl použít jako implementační prostředek jazyk **assembler**. Budu schopen lépe optimalizovat instrukce v časově kritických úsecích programu, než překladače vyšších programovacích jazyků.

Z generátoru hodinového signálu jsem podle aplikačního manuálu mikroprocesoru [4] spočítal frekvenci vnitřní sběrnice mikroprocesoru:

$$f_{BUS} = \frac{f_{OSC}}{4} = \frac{9830400}{4} = 2457600 \text{ [Hz]}.$$

Časové intervaly jsou měřené volně běžícím čítačem v jednotce TIM, frekvence hodinového signálu je totožná s f_{BUS} . Z pohledu implementace je užitečnější hodnota jeho periody:

$$T_{TIM} = \frac{1}{f_{BUS}} = \frac{1}{2457600} = 407 \text{ [ns]}.$$

Hodnota registru čítače vynásobeného T_{TIM} odpovídá skutečnému časovému úseku. Předchozí úvaha je klíčová pro samotnou implementaci a budu se na ní odakazovat.

4.1 RC5 protokol

Vysílač:

Odesílání dat zajišťuje podprogram `rc5_send`. Adresa zařízení a data podprogram získá z osmibitových globálních proměnných `addr` a `data`. Protokol RC5 definuje menší bitovou délku adresy a datům než je velikost proměnné. Užitečná data musí být umístěny na nižších bitech globálních proměnných. `Toggle` bit je součástí parametru `addr` a je před nejvyšším bitem adresy.

Algoritmus odesílání pracuje v cyklu, který se opakuje čtrnáctkrát, podle počtu bitu datového rámce protokolu. Před vstupem do cyklu se upraví proměnné `addr` a `data`, aby se později mohly odesílat jejich bity prostým logickým posunem. V iteraci cyklu dojde ke

zjištění logické hodnoty bitu a podle ní se generuje nejdříve značka nebo mezera. Oba stavy trvají polovinu bitové periody a pak se změní. Dokročení na bitovou půlperiodu způsobí přetečení čítače, který je nastaven jako volně běžící čítač do hodnoty odpovídající bitové půlperiodě. Tato hodnota je nastavená v registru čítače a vypočítá se vydělením skutečného času trvání půlperiody hodnotou T_{TIM} z úvodu kapitoly:

$$\frac{889 \mu s}{407 ns} \doteq 2184.$$

Při generování mezery je nastaven výstupní pin (nultý portu A) na logickou jedničku. Značka musí měnit na výstupním pinu logické hodnoty s modulační frekvencí, která je 36 KHz. Využívám k tomu nultý kanál čítače. Jeho komparační registr je nastaven na hodnotu odpovídající periodě modulační frekvence. Hodnota komparačního registru se spočte vydělením modulační periody konstantou T_{TIM} :

$$\frac{1}{\frac{36 KHz}{407 ns}} \doteq 68.$$

Jakmile volně běžící čítač dosáhne této hodnoty, vyvolá se přerušení nultého kanálu a neguje se výstupní pin. V obsluze přerušení se konstanta 68 přičítá ke komparačnímu registru po celou dobu trvání bitové půlperiody. Musí se provádět šestnáctibitové sečítání, protože registr je šestnáctibitový. Aby se generovala skutečně frekvence 36 KHz, musí být obsluha přerušení a všechny ostatní režijní instrukce provedeny za kratší dobu než 68 taktu mikroprocesoru. Zde je výhodou zvolený programovací jazyk, který umožňuje optimalizovat výpočet. Obrázek 4.1 ukazuje úryvek kódu obsluhy přerušení s optimalizovaným výpočtem. Čísla v komentářích za jednotlivými instrukcemi udávají počet taktů procesoru potřebných k jejich vykonání. Jejich součet je 30. Dalších deset taktů trvá přepnutí kontextu mikroprocesoru do obsluhy přerušení. Celkový počet režijních instrukcí je 40.

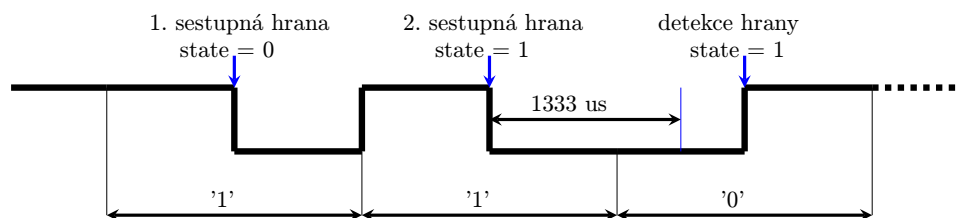
gen_period:			
BCLR 7,TSCO	; 4	<i>nulování příznaku přerušení</i>	
LDA TCHOL	; 3	<i>kopírování hodnot</i>	
LDX TCHOH	; 3		
ADD mark_ir	; 3	<i>přičtení půlperiody modulační frekvence</i>	
BCC not_inc_tch0h	; 3	<i>detekce přenosu</i>	
INCX	; 1	<i>přičtení přenosu</i>	
STX TCHOH	; 3	<i>aktualizování hodnoty registru</i>	
not_inc_tch0h:			
STA TCHOL	; 3	<i>aktualizování hodnoty registru</i>	
RTI	; 7	<i>návrat z přerušení</i>	

Obrázek 4.1: Úsek kódu v jazyce assembler generující modulační frekvenci

Přijímač:

Příjem datového rámce protokolu RC5 zabezpečuje podprogram `rc5_receive`. Nejedná se o klasický podprogram, ale o obsluhu přerušení periferní jednotky detekce hrany. Přerušení vyvolá sestupná hrana na přijímacím pinu a způsobí změnu stavu přijímače. Stav je uložen v globální proměnné `state`. Přijímač se může nacházet ve dvou stavech: detekce datového rámce a příjem bitů rámce. Po vykonání instrukcí v daném stavu mikroprocesor opouští obsluhu přerušení a čeká na další přerušení. Příjem neúplného nebo špatně časovaného

rámce způsobí přerušení vyvolané přetečením volně běžícího čítače. Proměnná `state` se vynuluje a tím se přijímač dostane do výchozího stavu. Algoritmus dekódování protokolu ilustruje obrázek 4.2. Datový rámec protokolu RC5 začíná dvěma start bity s hodnotou



Obrázek 4.2: Příjem RC5 protokolu

logické jedničky. První detekovaná sestupná hrana je tedy uprostřed bitové periody prvního start bitu. Přijímač nastaví do proměnné `bitcnt` celkový počet datových bitů protokolu, přesune se do druhého stavu a opouští kód obsluhy přerušení. Druhá detekovaná sestupná hrana je v bitové půlperiodě druhého start bitu. Přijímač se nachází ve druhém stavu a začíná provádět algoritmus určení logické hodnoty bitu. Nastaví volně běžící čítač na třičtvrtinu ($1333 \mu s$) bitové periody a čeká na jeho přetečení. Hodnota modulo registru čítače odpovídá podílu skutečného požadovaného času a konstanty T_{TIM} :

$$\frac{1333 \mu s}{407 ns} \doteq 3276.$$

Přetečení nastane ve čtvrtině bitové periody následujícího přijatého bitu. Provede se čtení výstupu demodulátoru (první pin portu A) a určení jeho logické hodnoty. Dále se zkontroluje počet přijatých bitů a pokud ještě není přijatý celý datový rámec nastaví se jednotka záchytu hrany na detekci libovolné hrany. Tím se zajistí vyvolání přerušení zase v polovině bitové periody následujícího bitu.

Přijatá data jsou umístěné v globálních proměnných `addr` a `data`.

4.2 NEC protokol

Vysílač:

Podprogram `nec_send` provede odeslání dat protokolem NEC. Data mu jsou předány jako osmibitové globální proměnné `data` a `addr`. Protokol vyžaduje odeslání dat i jejich negací. Celkem se tedy odesílá čtyři bajty. Všechny čtyři bajty se nejdříve umístí ve správném pořadí na zásobník. Algoritmus pracuje ve dvou zanořených cyklech. První vybírá data o velikosti jednoho bajtu ze zásobníku. Druhý, vnořený cyklus, tyto data odesílá bit po bitu, dokud neodešle celý bajt.

Kódování logických hodnot stanovuje stejně dlouhou značku u obou, umístěnou na začátku jejich bitových period. Hodnoty se rozlišují pouze délkou trvání následné mezery. Délka mezery je u logické nuly shodná s délkou značky, u logické jedničky trvá trojnásobně dlouho. Z předchozích úvah plyne, že pro generování kódovaných logických hodnot postačuje nastavit volně běžící čítač pouze jednou na konstantu odpovídající délce značky logických hodnot. Ostatní časy jsou její násobky. Ve výsledku pak vzniknou přesnější časové výstupy, než kdyby se před každým generováním značky nebo mezery nastavoval a nuloval čítač. Nepřesnosti by způsobovaly režijní instrukce.

Vnořený cyklus nejdříve generuje značku trvající $560 \mu s$. Pak zjišťuje hodnotu odesílaného bitu. Podle ní se nechá přetéct čítač jednou nebo třikrát. Konstanta modulo registru čítače pro časový úsek $560 \mu s$ odpovídá podílu této hodnoty konstantou T_{TIM} :

$$\frac{560 \mu s}{407 ns} \doteq 1376.$$

Jak již bylo zmíněno, kódování logických hodnot se liší pouze v délce mezery. Z kapitoly 3.2.1 víme, že přijímací modul interpretuje mezeru i klidový stav přenosového kanálu logickou jedničkou. Vzniká problém při odesílání posledního bitu. Aby ho bylo možno správně interpretovat na přijímací straně, vysílá se za ním ještě značka délky $560 \mu s$.

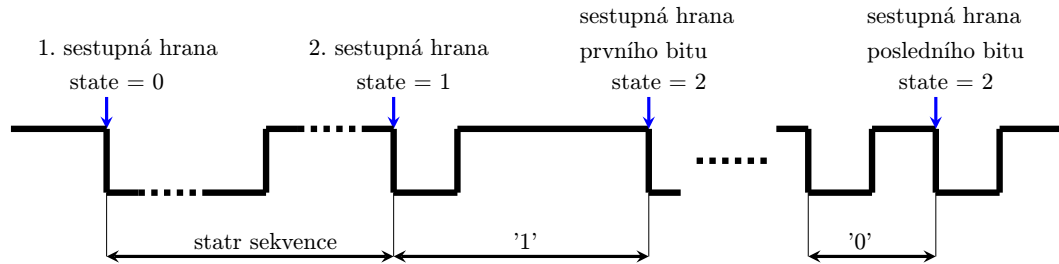
Značka o frekvenci $38 KHz$ se generuje stejným způsobem jako u protokolu RC5. Pouze komparační registr nultého kanálu TIM jednotky obsahuje nižší konstantu. Vypočítá se stejným vzorcem, ale s dosazenou frekvencí $38 KHz$. Výsledná konstanta je 65.

Před odesíláním dat se odesílá start sekvence o délce $9 ms$ značky a $2, 25 ms$ mezery.

Přijímač:

Příjem rámců NEC protokolu začíná detekcí sestupné hrany na pinu připojeném k přijímacímu modulu. Generuje se přerušení TIM jednotky, kanálu 1. Obsluhu přerušení zajišťuje úsek kódu s návěštím `nec_receive`. Stejně jako u přijímače RC5 protokolu se mezi detekovanými přerušeními spouští čítač za účelem zjištění chybného příjmu.

Pracuje se ve třech stavech, číslováných od nuly: detekce příjmu, kontrola start sekvence a příjem bitů. Každý provádí svůj kód, na konci přejde do vyššího stavu a opouští přerušení. Nultý provádí inicializaci a je současně výchozí. Nastavuje proměnnou `bitcnt` na počet přijímaných bitů, celkem 32. První stav kontroluje správnou délku start sekvence. Druhý určuje hodnoty přijatých bitů a ukládá je do paměti. Určování hodnoty, stejně jako kontrola délky start sekvence, spočívá v porovnávání časových úseků mezi přijatými sestupnými hranami s časy stanovenými protokolem NEC. Situaci ilustruje obrázek 4.1.



Obrázek 4.3: Příjem NEC protokolu

Změřené časy se uloží do komparačního registru prvního kanálu TIM jednotky, po detekované hraně. Hodnota je šestnáctibitová, uložena ve dvou osmibitových registrech. Musí se počítat s průměrnou odchylkou těchto hodnot, jinak by minimální odchylka znehodnotila příjem. S ohledem na tyto skutečnosti jsem se rozhodl porovnávat pouze hodnotu horního registru s vhodnou konstantou. Konstanta představuje očekávaný čas trvání start sekvence nebo jedné z bitových period. Jedná se zase o horní bajt šestnáctibitových hodnoty. Zanedbáním dolních osmi bitů vznikne tolerance přijímané hodnoty o absolutní velikosti druhé mocniny osmi. Pokud je ovšem číselná hodnota zanedbaného bajtu konstanty příliš nízká (vysoká), sníží se tolerance nižších (vyšších) naměřených časů. Problém

jsem vyřešil porovnáváním s další konstantou, která je volená menší (větší) o jedničku při nízké (vysoké) hodnotě zanedbávaného bajtu. Toto řešení sníží počet komparačních operací a stanoví dostatečný toleranční interval naměřených časů. Interval je dán druhou mocninou osmi nebo devíti, podle počtu porovnávacích konstant. Konkrétní zvolené konstanty, očekávané hodnoty stanovené protokolem a tolerované mezní hodnoty zachycuje tabulka 4.1. Experimentálně jsem provedl několik měření vysílače NEC protokolu s použitím platformy FitKit a zachytil průměrnou odchylku 15 taktů. Intervaly přijímače jsou tedy navržené s dostatečnou tolerancí.

Příjem nesprávného časového průběhu se projeví neshodou s žádnou z konstant. Přijímač přejde do stavu nula, kde čeká na nový datový rámec. Korektní příjem nastaví bit IR příjmu v proměnné **status**.

	jednotka	start sekvence	log. 1	log. 0
očekávaná hodnota	μs	11250	2240	1120
	takty*	6BF9	157F	ABF
porovnávací konstanta 1	takty*	6B	15	A
porovnávací konstanta 2	takty*	6C	-	B
min. tolerovaná hodnota	μs	11149	2188	1042
	takty*	6B00	1500	A00
max. tolerovaná hodnota	μs	11357	2292	1250
	takty*	6CFF	15FF	BFF

Tabulka 4.1: Zvolená tolerance přijímaných časových průběhu NEC protokolu. * jednotka takty značí počet hodinových pulsů čítače a její hodnota je v šestnáctkové číselné soustavě

4.3 SIRC protokol

Vysílač:

Stejně jako u vysílačů předchozích protokolů se i u vysílače SIRC protokolu data a adresa předávají globálními proměnnými **data** a **addr**. Podprogram pro odeslání dat má název **sirc_send**.

Výhodou časování tohoto protokolu je základní časová konstanta $T_{SIRC} = 600 \mu s$, jejíž násobky lze použít pro generování start sekvence i logických hodnot. Odpovídající hodnota volně běžícího čítače je podíl T_{SIRC} a konstanty T_{TIM} :

$$\frac{600 \mu s}{407 ns} \doteq 1447.$$

Jeho přetečení signalizuje uplynutí časové konstanty T_{SIRC} .

Logické hodnoty začínají značkou a liší se délkou jejího trvání. Interpretace značky na přijímači je jiná než stav přijímače, kdy se nic neodesílá. Z těchto důvodů není potřeba generovat zakončující značku za posledním přeneseným bitem jako u NEC protokolu.

Algoritmus odesílání dat pracuje v cyklu opakujícím se podle počtu bitu datového rámce. Bity se odesílají od nejméně významného datového. Délka trvání značky a mezery v bitové periodě je kontrolována počtem přetečení čítače, podle logické hodnoty bitu.

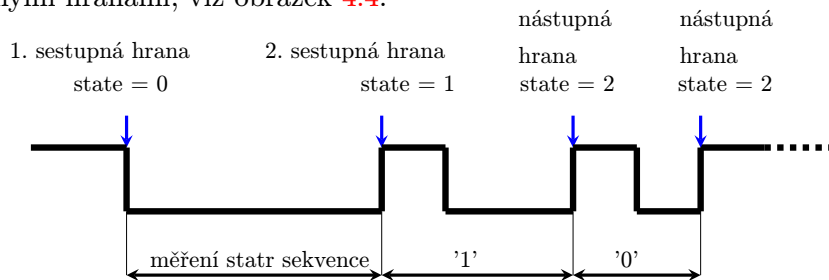
Generování značky frekvence $40 KHz$ zajišťuje, stejně jako u implementace předchozích protokolů přerušení TIM jednotky nultého kanálu. Konstanta odpovídající této frekvenci se

v přerušení přičítá k hodnotě komparačního registru a neguje se výstupní nultý pin portu A. Konstanta je dána podílem periody frekvence 40 KHz a konstanty T_{TIM} :

$$\frac{1}{\frac{40\text{ KHz}}{407\text{ ns}}} \doteq 61.$$

Přijímač:

Podobně jako předchozí přijímače i tento využívá jednotku TIM a její zdroje přerušení. Část kódu, která řeší příjem začíná návěštím `sirc_receive`. Realizuje obsluhu přerušení jednotky detekce hrany. Přijímač se může nacházet ve třech stavech: detekce příjmu, kontrola správné délky start sekvence a dekodování bitů. Přechody mezi nimi jsou dány detekovanými hranami, viz obrázek 4.4.



Obrázek 4.4: Příjem SIRC protokolu

V prvním stavu kontroly periody start sekvence se měří čas mezi první sestupnou a nástupnou hranou. Ve druhém stavu dekodování bitů se měří čas do následující náběžné hrany. Pokud je čas mezi detekovanými hranami příliš dlouhý, dojde k přetečení volně běžícího čítače a přechodu do stavu detekce příjmu. Neúspěšné dekodování bitů nebo chybná perioda start sekvence také způsobí přechod do výchozího nultého stavu. Úspěšné přijetí datového rámce nastaví bit IR příjmu v proměnné `status`.

Změřené časy se porovnávají s časy stanovenými protokolem SIRC. Porovnávání musí probíhat s určitou tolerancí. Řešení komparace je stejné jako u přijímače protokolu NEC, kapitola 4.2. Změřené hodnoty jsou šestnáctibitové ale porovnávání probíhá pouze s horním bajtem. Porovnávací konstanty odpovídají časům stanoveným protokolem. Přepoččet je prosté vydělení požadované časové konstanty konstantou T_{TIM} . Výsledná hodnota je interpretována jako šestnáctibitová. Porovnávací konstanta obsahuje pouze hodnotu jejího horního bajtu. Při malých nebo velkých hodnotách zanedbaného spodního bajtu se zmenšuje tolerance a musí se porovnávat s konstantou o jedničku menší nebo větší, přesnější zdůvodnění viz přijímač protokolu NEC v kapitole 4.2. Očekávané hodnoty stanovené protokolem, od nich odvozené porovnávací konstanty, minimální a maximální tolerované hodnoty jsou v tabulce 4.2.

Podobně jako u protokolu NEC, jsem experimentálně změřil časy vysílače SIRC a získal průměrnou odchylku časových průběhu o velikosti $35\text{ }\mu\text{s}$. Zvolená tolerance přijímače je tedy dostačující.

	jednotka	start sekvence	log. 1	log. 0
očekávaná hodnota	μs	2400	1800	1200
	takty*	1708	1146	B84
porovnávací konstanta 1	takty*	16	10	B
porovnávací konstanta 2	takty*	17	11	-
min. tolerovaná hodnota	μs	2292	1667	1146
	takty*	1600	1000	B00
max. tolerovaná hodnota	μs	2500	1875	1250
	takty*	17FF	11FF	BFF

Tabulka 4.2: Zvolená tolerance přijímaných časových průběhu SIRC protokolu. * jednotka takty značí počet hodinových pulsů čítače a její hodnota je v šestnáctkové číselné soustavě

4.4 IrDA přenos

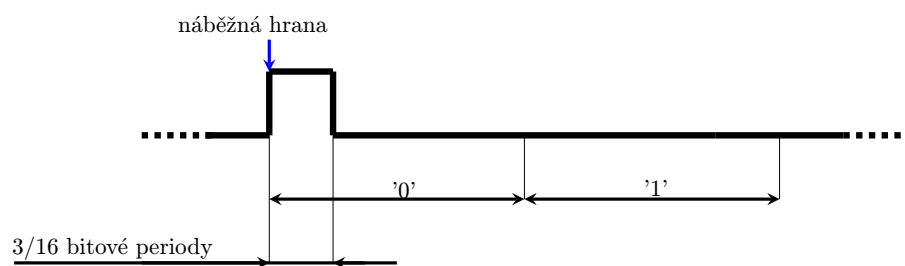
Přenos dat definovaný standardem IrDA odpovídá asynchronnímu přenosu UART s pulsní modulací bitů, viz kapitola 2.2. Při implementaci jsem vycházel z programu, který emuluje UART přenos na mikroprocesoru Nitron. Program je k dispozici jako součást dokumentace k vývojovému kitu Janus. Zachoval jsem stejné příznakové bity přenosu, přidal pulsní modulaci přenášených bitů a vyhradil další pin mikroprocesoru, aby příjem a odesílání dat nesdílelo společný.

Přenášený datový rámec je desetibitový a skládá se ze start bitu, osmi datových bitů a jednoho stop bitu. Přenosovou rychlost lze nastavit změnou konstant `BITTIME`, `BITHTIME` a `BITTIME16`. Jejich význam je následující: bitová perioda v taktech, polovina bitové periody v taktech a šestnáctina bitové periody v taktech. Standard IrDA pro protokol SIR definuje minimální rychlost 2,4 Kb/s a maximální 115,2 Kb/s. S ohledem na taktovací frekvenci použitého mikroprocesoru lze dosáhnout maximální přenosové rychlosti 19,2 Kb/s. Konkrétní hodnoty konstant odpovídající přenosovým rychlostem jsou uvedeny v kapitole 6, dokumentace.

Vysílač pracuje ve dvou stavech. První odešle start bit a nastaví čítač na hodnotu bitové periody. Druhý stav provádí odesílání osmi datových bitů a stop bitu. Odeslání bitu probíhá generováním vysoké úrovně na výstupním pinu mikroprocesoru po dobu 3/16 bitové periody u logické nuly nebo generování nízké úrovně na výstupním pinu po celou dobu bitové periody u logické jedničky. Doba trvání vysoké úrovně odměřuje zpožďovací smyčka, která provádí dekrementaci hodnoty konstanty `BITTIME16` k nule. Konstanta `BITTIME16` odpovídá 3/16 bitové periody zvolené přenosové rychlosti, přepočtené na takty mikroprocesoru a vydělené třemi. Tři je doba trvání jedné iterace zpožďovací smyčky v taktech.

Přijímač je implementován jako obsluha přerušení jednotky detekce hrany, konkrétně nultého kanálu periférie TIM a obsluhy přerušení přetečení časovače. První přerušení od jednotky detekce hrany signalizuje start bit přijímaného datového rámce. Následně přijímač čeká polovinu bitové periody, pak nastaví hodnotu modulo registru čítač na celou bitovou periodu a opustí kód tohoto přerušení. Přetečení čítače nastane v polovině periody prvního datového bitu. Zde se zkontroluje zda během této doby byl nastaven příznak detekované náběžné hrany v nultém kanálu jednotky TIM, který je směrodatný pro určení logické hodnoty přijímaného bitu. Nekontroluji tedy délku možného pulsu odpovídajícího zakódované logické nule, pouze jeho přítomnost. Pokud není nastavený příznak detekce náběžné hrany, hodnota bitu se považuje za logickou jedničku. Stop bit je dekodován stejným způsobem a

pokud jeho hodnota neodpovídá logické jedničce, je celý přijatý bajt zahozen jako chybný. Výstupní signál, z obvodu HSDL-3612, je znázorněn na obrázku 4.5.



Obrázek 4.5: Výstup generovaný obvodem HSDL-3612

Kapitola 5

Ukázková aplikace

Zařízení, využívající protokoly dálkových ovladačů, často obsahují pouze vysílač nebo přijímač. Prvním cílem ukázkové aplikace je navrhnout univerzální modul tak, aby byl schopen přijímat i vysílat pomocí zvoleného protokolu. Druhým cílem je vytvořit modul jako periferní zařízení typu **slave** (podřízené) s dalším rozhraním. Toto rozhraní bude přeposílat data z IR rozhraní k zařízení typu **master** (nadržené) a naopak. Principiální funkce modulu tedy bude převodník IR protokolu na protokol jiného rozhraní.

5.1 Popis a řešení ukázkové aplikace

Aplikace demonstruje vzorové použití modulů a funkcí, navržených v této práci, ke komunikaci dvou mikroprocesorů. Blokové schéma navrhované aplikace ukazuje obrázek 5.1. Mikroprocesory uP 1 a uP 2 mají připojené terminály, které umožňující uživateli odeslat, přijmout a zobrazit data. Při praktickém využití lze jednoduše připojit další senzory a čidla. Data mikroprocesory odesílají pomocí SPI rozhraní IR modulům. Zvolený přenosový protokol SPI rozhraní bude popsán v následující podkapitole.

Komponenty modul 1 a modul 2 zajišťují IR komunikaci. Realizují převod SPI rozhraní na IR protokoly NEC, SIRC nebo RC5. Konkrétní protokol se volí konfigurací pinů a je nezávislý na SPI rozhraní.

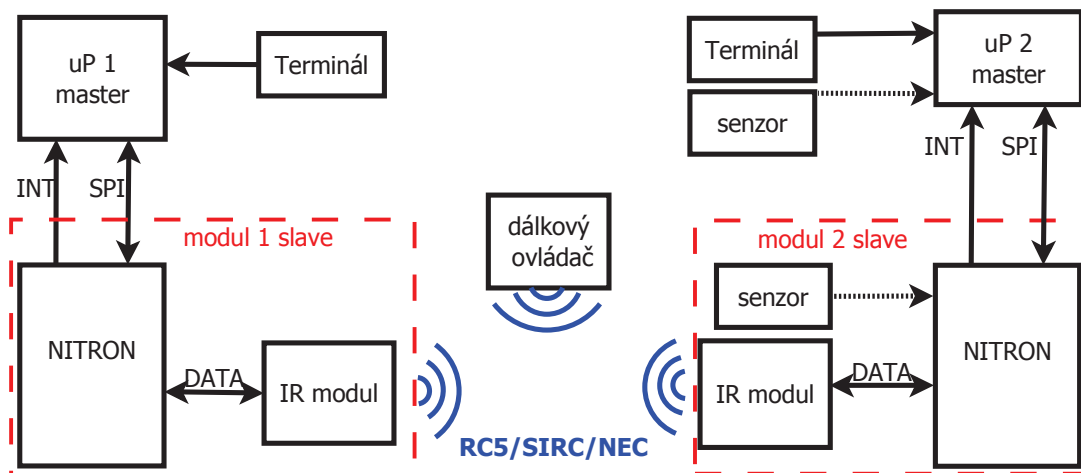
Logika modulu nedovoluje současnou komunikaci na obou rozhraních. Přenosy na nich jsou zcela atomické a nepřerušitelné. Po dokončení příjmu na libovolném rozhraní, se data ihned odesílají druhým rozhraním.

Data je možno samozřejmě odesílat IR modulům i pomocí dálkového ovladače konkrétního IR protokolu.

5.1.1 Volba druhého rozhraní modulu

Nabízí se několik standardních rozhraní: I2C, SPI, CAN, UART, a další. Aby se maximálně šetřil strojový čas nadřazeného mikroprocesoru, musí sám mít fyzickou podporu voleného rozhraní. Na druhou stranu mikroprocesor Nitron, který řídí modul, nenabízí fyzickou podporu žádného z uvedených protokolů. Programové řešení rozhraní by mělo být co nejjednodušší.

Použil jsem rozhraní SPI ve verzi s jedním datovým vodičem. Programové řešení SPI rozhraní na jediném procesoru totiž neumožňuje plně duplexní komunikaci. Podřízené zařízení žádá o odeslání dat nadřazenému změnou logické hodnoty na dalším použitém vodiči. Nadřízený mikroprocesor nesmí začít komunikaci dokud není připravený podřízený. Stav



Obrázek 5.1: Blokové schéma modulu ukázkové aplikace

připravenosti indikuje datový vodič ve stavu logické jedničky. Na obrázku 5.1 je vidět blokové schéma ukázkové aplikace a využití jednotlivých rozhraní.

5.1.2 Návrh protokolu SPI rozhraní a implementace v HC08

Rozhraní SPI definuje přenos bajtů mezi zařízeními typu master a slave. Komunikaci zahájí vždy nadřízené zařízení generováním hodinového signálu. Při klasickém řešení se data přenášejí mezi oběma po samostatných vodičích. Data jsou přijímána (odesílána) s hranou hodinového signálu do (z) posuvného registru.

Jediný vodič pro data vyžaduje odeslání režijního bajtu nadřízeným mikroprocesorem, s informací o směru komunikace. Podřízený mikroprocesor přečte první bajt a poté odesílá nebo přijímá. Data mají velikost dva bajty a tvoří je adresa vzdáleného IR zařízení a data jemu určené.

Programová implementace SPI rozhraní je založena na určování hodnot datového pinu, při náběžné hraně pinu hodinového signálu. Přenos zahájí první sestupná hrana pinu hodinového signálu, která vyvolá vnější přerušení IRQ. Obsluha přerušení volá podprogram `spi_receive` a přijme první bajt. Určí směr přenosu a voláním stejného podprogramu přijme další dva bajty nebo volá podprogram `spi_send` a odesílá dva bajty. Během komunikace jsou zakázány všechny zdroje přerušení. Oba podprogramy přepínají datový pin mezi vstupním a výstupním módem.

5.2 Celková implementace aplikace

Odesílání dat moduly zajišťuje podprogram `ir_send`. Ve svém těle vybírá IR protokol podle šestého a sedmého pinu portu B. Podobné rozhodování proběhne po detekci spádové hrany na přijímacím pinu. Úspěšný příjem dat nastaví příznakové bity v globální proměnné

status. Atomičnost a nepřerušitelnost těchto operací zajišťuje vhodné povolování a zákazy přerušení.

Vývojový diagram řídicího programu mikroprocesoru Nitron je v příloze **B.1**.

Nadřízené mikroprocesory pracují v cyklu a čekají na příjem nebo pokyn k odeslání z terminálu. Data jsou zobrazována na LCD displeji. Jako implementační jazyk nadřízených mikroprocesorů jsem zvolil jazyk c.

Vývojový diagram řídicího programu Fitkitu je v příloze **B.2**. Řídicí program mikroprocesoru Atmel je podobný, jediná změna je v používání dipů místo klávesnice.

5.3 Zapojení

Moduly zabezpečující IR přenos jsou totožné s navrženými v této práci. Jako nadřízené zařízení jsem použil platformu Fitkit a mikroprocesor firmy Atmel At89s8253. Oba mají připojené LCD displeje. Fitkit ovládá přenos maticovou klávesnicí. Mikroprocesor Atmel lze ovládat osmibitovým dipem připojeným k jednomu z jeho portu.

5.4 Použití aplikace

Prostřednictvím terminálů lze odesílat, přijímat a zobrazovat data. Pomocí klávesnice platformy Fitkit se vybírají data odpovídající binárním číslům kláves. Klávesou označenou písmenem A se data odešlou. Mikroprocesor Atmel volí data o hodnotě nastavené na sedmi bitech dipu. Nastavením osmého bitu dipu dojde k odeslání.

Moduly disponují také dipy pro snadnou volbu IR protokolu, podle tabulky **5.1**.

protokol	pin 7 portu B	pin 6 portu B
RC5	0	0
NEC	0	1
SIRC	1	0

Tabulka 5.1: Volba IR protokolů v ukázkové aplikaci

Fotografie zapojení testovací aplikace na nepájivém poli je v příloze **C.1**. Tato fotografie byla pořízena v situaci, kdy mikroprocesor Atmel odeslal data o hodnotě 4 na adresu o hodnotě 1. LCD Atmelu zobrazuje na druhém řádku v binární podobě odeslanou adresu a hned za ní data. LCD fitkitu zobrazuje přijaté data a adresu stejným způsobem na prvním řádku. Hodnota dipů připojených k Atmelu odpovídá binární hodnotě přenášených dat, čísla 4. Osmý dip je sice nastaven, ale slouží pro iniciování přenosu. Na hodnotu odesílaných dat nemá vliv. Přenosový protokol je zvolen na dipech mikroprocesorů Nitron, jako RC5 protokol.

Kapitola 6

Dokumentace

6.1 Podprogramy protokolů dálkových ovladačů

Podprogramy protokolů dálkových ovladačů jsou umístěné v souboru `ir.inc` a mají totožné parametry. Liší se v názvu, kde každý začíná názvem protokolu a pokračuje `_send` nebo `_receive`. Všechny jsou nepřerušitelné. Po úspěšném příjmu nastaví sedmý bit globální proměnné `status`.

Návěští končící řetězcem `_receive` nejsou podprogramy, ale obsluhy přerušení. Nesmí se k nim přistupovat instrukcemi pro volání podprogramů. Provádí se na ně skok pouze z obsluhy přerušení TIM jednotky, prvního kanálu. Nenulují příznak přerušení. Poslední instrukce jejich kódu je návrat z obsluhy přerušení. V případě korektního příjmu dat je nastaven sedmý bit globální proměnné `status`.

Parametry podprogramů jsou osmibitové globální proměnné `addr` a `data`. Některé protokoly stanovují data menší bitové šířky. V takových případech jsou za nadbytečné bity považovány horní a neodesílají se. Přijatá data mají nadbytečné bity nulové.

Omezení: Podprogramy využívají TIM jednotku architektury HC08.

6.2 Podprogramy standardu IrDA

Soubor `irda.inc` obsahuje podprogramy pro komunikaci standardem IrDA. Realizují přenos rychlostmi podle zvolených hodnot konstant `BITTIME`, `BITHTIME` a `BITTIME16`. Konkrétní hodnoty ukazuje tabulka 6.1.

rychlost v Kb/s	konstanty v taktech		
	BITTIME	BITHTIME	BITTIME16
2,4	1024	512	64
4,8	512	256	32
9,6	256	128	16
19,2	128	64	8

Tabulka 6.1: Hodnoty konstant odpovídající přenosovým rychlostem

Podprogram `uart_init` nastaví patřičné periférie procesoru na příjem dat. Podprogram `tx_char` odešle data uložená v akumulátoru. Příjem dat je indikován nastavením čtvrtého bitu v globální proměnné `trxstat`. Data jsou pak dostupné v proměnné `rxdata`.

Omezení: Podprogramy využívají TIM jednotku architektury HC08.

Kapitola 7

Závěr

Cílem práce bylo navrhnout hardwarové moduly a jejich ovládací firmware, které budou zabezpečovat spolehlivý přenos dat v oblasti infračerveného záření na platformě Freescale HC08. Zaměřil jsem se na protokoly dálkových ovladačů NEC, SIRC, RC5 a standard IrDA.

Z několika návrhů modulů jsem nakonec vytvořil dva, které jsou schopné komunikovat stanovenými protokoly. Jeden pro protokoly dálkových ovladačů a druhý pro standard IrDA. Doručení datového rámce mezi dvěma moduly bylo spolehlivé do vzdálenosti 3,5 metrů u modulů dálkových ovladačů. Vyšší dosah by bylo možné dosáhnout snížením hodnoty ochranného odporu a tím zvýšit proud vysílací LED diody. Při přiměřeném snížení odporu by nemělo dojít k jejímu zničení, protože je spínaná pulsně. Podobě lze navýšit přenosovou vzdálenost bez chyb přenosu u modulu pro IrDA komunikaci, kterou jsem naměřil 1,3 metrů při dodržení úhlu ne většího než 30° mezi vysílačem a přijímačem.

Firmware modulu pro IrDA komunikaci umí vysílat maximálně s rychlostí 115,2 Kb/s. Trojnásobné rychlosti by bylo možné dosáhnout použitím krystalu s kmitočtem 32 MHz, který je maximálně přípustný pro zvolený mikroprocesor.

Použití navržených modulů jsem demonstroval v ukázkové aplikaci, která realizuje převodník mezi protokoly dálkových ovladačů a SPI rozhraním. Takovéto zařízení půjde uplatnit všude, kde je potřeba asynchronní komunikace s jinými vzdálenými zařízeními, která disponují pouze SPI rozhraním. Například sběr dat s několika čidel. Nevýhodou je nutnost přímé viditelnosti mezi účastníky komunikace a kolize při současné komunikaci více zařízení. Druhý problém lze odstranit zavedením vhodného komunikačního protokolu s mechanismem potvrzování dat.

Firmware ukázkové aplikace ve zvoleném mikroprocesoru zabral necelé 3 KB flash paměti programu. Zbylou paměť o velikosti přibližně 900 B by bylo možné v budoucnu využít pro implementaci datových front pro SPI a IR rozhraní.

Literatura

- [1] Bergmans, S.: *NEC Protocol*. [online], 2011-05-29 [cit. 2012-01-20].
URL <http://sbprojects.com/knowledge/ir/nec.php>
- [2] Bergmans, S.: *Philips RC-5 Protocol*. [online], 2011-05-29 [cit. 2012-01-20].
URL <http://sbprojects.com/knowledge/ir/rc5.php>
- [3] Bergmans, S.: *Sony SIRC Protocol*. [online], 2011-05-29 [cit. 2012-01-20].
URL <http://sbprojects.com/knowledge/ir/sirc.php>
- [4] Freescale Semiconductor: *MC68HC908QY4*. [online]. Katalogový list, Freescale Semiconductor, 2010 [cit. 2012-02-14].
URL http://cache.freescale.com/files/microcontrollers/doc/%data_sheet/MC68HC908QY4.pdf?fpsp=1
- [5] GM Electronic: *Infrared remote control receiver module*. [online]. Katalogový list, HB Electronic Components, 2004 [cit. 2012-02-16].
URL <http://www.gme.cz/dokumentace/511/511-575/dsh.511-575.%1.pdf>
- [6] Infrared Data Association: *Infrared Data Communication According the IrDA Standard, Part 1: Physical Layer*. [online]. Technická zpráva, Vishay Semiconductors, 2006 [cit. 2012-02-14].
URL <http://irda.org/associations/2494/files/Publications/P%hysical%20Basics.pdf>
- [7] Infrared Data Association: *Infrared Data Communication According the IrDA Standard, Part 2: Protocol*. [online]. Technická zpráva, Vishay Semiconductors, 2006 [cit. 2012-02-14].
URL <http://irda.org/associations/2494/files/Publications/Protocol%20Basics.pdf>
- [8] Infrared Data Association: *The secure Wireless Link*. [online], 2011-05-29 [cit. 2012-02-14].
URL <http://irda.org>
- [9] Martinek, R.: *Senzory v průmyslové praxi*. 190 stran, Nakladatelství BEN - technická literatura, první vydání, 2004, iISBN 80-7300-114-4.
- [10] Motorola: *CPU08 Central Processor Unit*. [PDF]. Referenční manuál, Motorola, 2001.
- [11] Náprstek, J.: *Normy pro IR datový přenos*. [online], 2002-08-31 [cit. 2012-02-16].
URL <http://www.hw.cz/produkty/nove-soucastky/nitron-jak-na-nej.html>

- [12] Picprojects: *Sony SIRC infrared protocol*. [online], 2010-02-10 [cit. 2012-01-21].
URL <http://picprojects.org.uk/projects/sirc/sonysirc.pdf>
- [13] Šára, Z.: *Normy pro IR datový přenos*. [online], 1999-03-4 [cit. 2012-01-20].
URL http://www.hw.cz/docs/ir_prenos/ir_protokoly.html
- [14] *GM Electronic: HSDL-3612*. [online]. Katalogový list, Agilent Technologies, 2003 [cit. 2012-02-16].
URL <http://www.gme.cz/dokumentace/968/968-009/dsh.968-009.%1.pdf>
- [15] Uhdeová, N.: *Fyzikální praktikum*. Skripta 120 stran, VUT Brno, Fakulta elektrotechniky a komunikačních technologií, 2003.
- [16] Zembol, F.: *Analýza parametru infracerveného zobrazení*. Bakalářská práce, VUT Brno, Fakulta elektrotechniky a komunikačních technologií, 2009/2010.

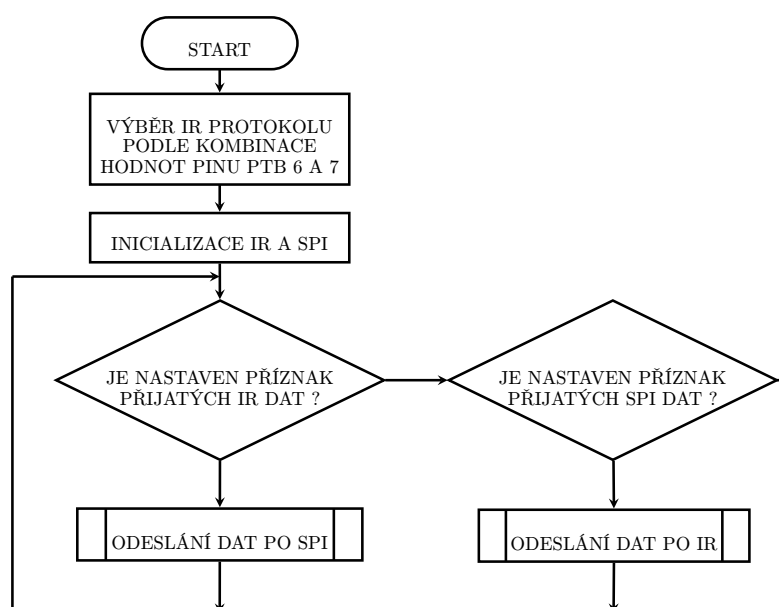
Příloha A

Obsah CD

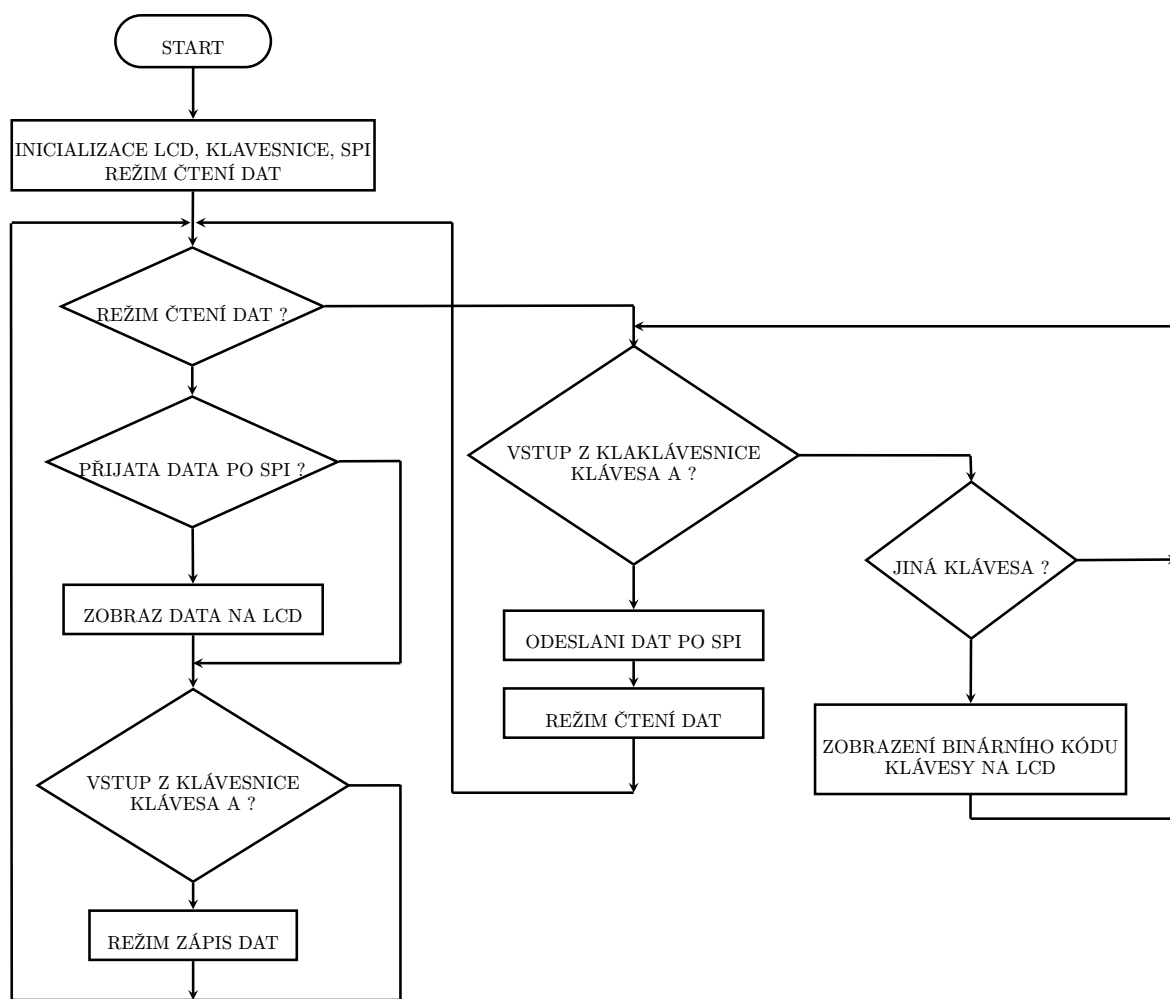
- `aplikace_ir.asm` ukázková aplikace
- `ir.inc` podprogramy pro ir komunikaci protokoly NEC, SIRC a RC5
- `irda.asm` testovací aplikace IrDA přenosu
- `irda.inc` podprogramy komunikující standardem IrDA
- `fitkit` adresář se soubory ovládacího program ukázkové aplikace na Fitkitu
- `atmel` adresář se soubory ovládacího programu ukázkové aplikace na mikroprocesoru Atmel
- `tex` adresář se zdrojovými kódy textu práce
- `projekt.pdf` text práce ve formátu pdf
- `test` adresář s pomocnými testovacími programy pro Fitkit

Příloha B

Vývojové diagramy ukázkové aplikace



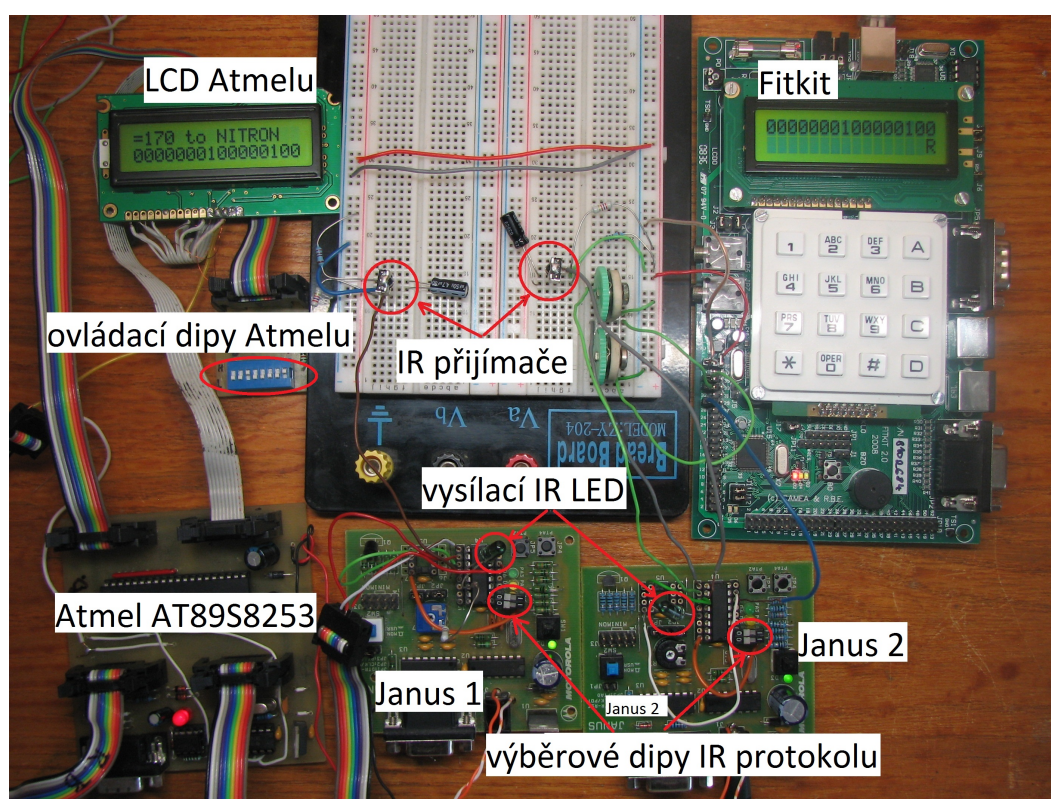
Obrázek B.1: Vývojový diagram řídicího programu Nitronu v ukázkové aplikaci



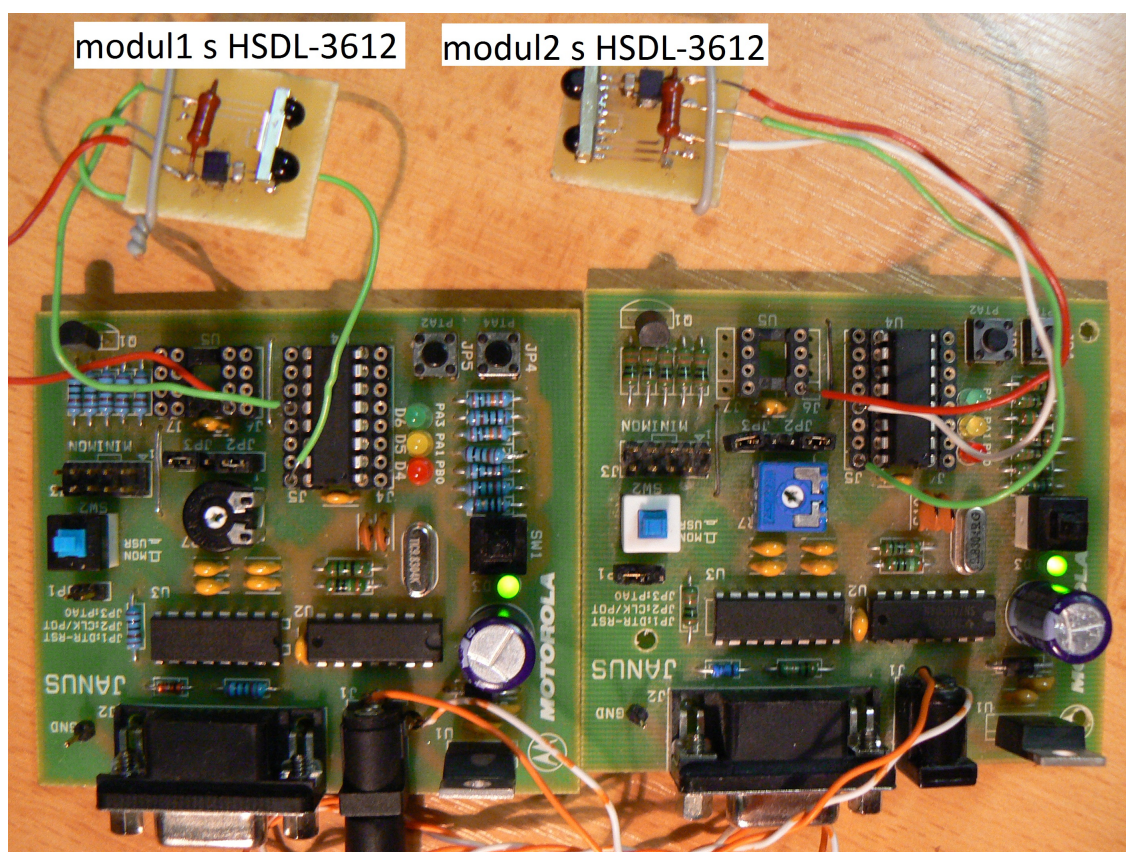
Obrázek B.2: Vývojový diagram řídicího programu Fitkitu v ukázkové aplikaci

Příloha C

Fotografie praktické části



Obrázek C.1: Zapojení ukázkové aplikace



Obrázek C.2: Testování modulu pro IrDA komunikaci